

Python: module browser.gui_functions

browser.gui_functions

[index](#)

```
# The PCMDI Data Browser Miscellaneous Functions - gui_functions module
#
#####
# Module:      gui_functions module
#
# Copyright:   "See file Legal.htm for copyright information."
#
# Authors:     PCMDI Software Team
#               Lawrence Livermore National Laboratory:
#               support@pcmdi.llnl.gov
#
# Description: PCMDI Software System Browser miscellaneous GUI functions.
#
# Version:    4.0
#
#####
# NOTE: need to use version of Python that imports Tkinter and Pmw
#-----
```

Modules

Numeric	copy_reg	math	sys
Tkinter	gui_support.gui_color	multiarray	types
_main	browser.gui_control	os	vcs
cdms	browser.gui_defined_variables	vcs.pagegui	
cdtime	browser.gui_message	pickle	
copy	browser.gui_reset	string	

Classes

[generate_labels_list](#)
[replace_axis_values](#)

class [generate_labels_list](#)

Methods defined here:

[__init__\(self, parent, dim_index, axis\)](#)

```
    generateList(self, button)
```

class **replace_axis_values**

Methods defined here:

```
    __init__(self, parent, dim_index, replace_type=None)
```

```
    evt_replace_axis_values(self, parent, dim_index, replace_type, result)
```

Functions

arange(...)

```
    arange(start, stop=None, step=1, typecode=None)
```

Just like range() except it returns an array whose type can be specified by the keyword argument typecode.

array(...)

```
    array(sequence, typecode=None, copy=1, saveSpace=0) will return a
```

arrayrange = arange(...)

```
    arange(start, stop=None, step=1, typecode=None)
```

Just like range() except it returns an array whose type can be specified by the keyword argument typecode.

choose(...)

```
    choose(a, (b1, b2, ...))
```

cross_correlate(...)

```
    cross correlate(a, v, mode=0)
```

evt_menu_index(parent, dim_index, called_from)

evt_menu_raw(parent, dim_index)

evt_transpose_dimensions(index, dim_name, t_index, t_dim_name, parent)

```
    #-----  
    # Transpose two dimensions  
    #-----
```

fromstring(...)

```
    fromstring(string, typecode='l', count=-1) returns a new 1d array
```

get_available_printers()

```
    #-----  
    # Return the list of available printers. Printer names should be 1  
    # in the user's $HOME/PCMDI_GRAPHICS/HARD_COPY file
```

```

#-----

get_axis_values(parent, dim_index)

get_axis_weight_values(parent, dim_index)

get_dimension_selections(Csrl, Cpts)
    #-----
    # Select the first and last dimension values -- Will also highlight
    # new range
    #-----


reshape(...)
    reshape(a, (d1, d2, ..., dn)). Change the shape of a to be an n-d array

return_axis_size(parent, dim_index)

return_latitude_region_values(region)

return_longitude_region_values(region)

searchsorted = binarysearch(...)
    binarysearch(a, v)

set_lat_lon(parent, R0, R1)

set_region_index_values(parent, dim_name, region, dim_index)

take(...)
    take(a, indices, axis=0). Selects the elements in indices from array a

zeros(...)
    zeros((d1, ..., dn), typecode='l', saveSpace=0) will return a new array

```

Data

Character = 'c'
Complex = 'D'
Complex0 = 'F'
Complex16 = 'F'
Complex32 = 'F'
Complex64 = 'D'
Complex8 = 'F'
Float = 'd'
Float0 = 'f'
Float16 = 'f'
Float32 = 'f'
Float64 = 'd'
Float8 = 'f'
Int = 'I'
Int0 = 'I'

```
Int16 = 's'
Int32 = 'i'
Int8 = '1'
LittleEndian = True
NewAxis = None
Pmw = <Pmw.Pmw_1_2.lib.PmwLoader.PmwLoader instance>
PyObject = 'O'
UInt = 'u'
UInt16 = 'w'
UInt32 = 'u'
UInt8 = 'b'
UnsignedInt16 = 'w'
UnsignedInt32 = 'u'
UnsignedInt8 = 'b'
UnsignedInteger = 'u'
absolute = <ufunc 'absolute'>
add = <ufunc 'add'>
arccos = <ufunc 'arccos'>
arccosh = <ufunc 'arccosh'>
arcsin = <ufunc 'arcsin'>
arcsinh = <ufunc 'arcsinh'>
arctan = <ufunc 'arctan'>
arctan2 = <ufunc 'arctan2'>
arctanh = <ufunc 'arctanh'>
bitwise_and = <ufunc 'bitwise_and'>
bitwise_or = <ufunc 'bitwise_or'>
bitwise_xor = <ufunc 'bitwise_xor'>
ceil = <ufunc 'ceil'>
conjugate = <ufunc 'conjugate'>
cos = <ufunc 'cos'>
cosh = <ufunc 'cosh'>
divide = <ufunc 'divide'>
divide_safe = <ufunc 'divide_safe'>
e = 2.7182818284590451
equal = <ufunc 'equal'>
exp = <ufunc 'exp'>
fabs = <ufunc 'fabs'>
floor = <ufunc 'floor'>
floor_divide = <ufunc 'floor_divide'>
fmod = <ufunc 'fmod'>
greater = <ufunc 'greater'>
greater_equal = <ufunc 'greater_equal'>
hypot = <ufunc 'hypot'>
invert = <ufunc 'invert'>
left_shift = <ufunc 'left_shift'>
less = <ufunc 'less'>
less_equal = <ufunc 'less_equal'>
log = <ufunc 'log'>
log10 = <ufunc 'log10'>
logical_and = <ufunc 'logical_and'>
logical_not = <ufunc 'logical_not'>
logical_or = <ufunc 'logical_or'>
```

```
logical_xor = <ufunc 'logical_xor'>
maximum = <ufunc 'maximum'>
minimum = <ufunc 'minimum'>
multiply = <ufunc 'multiply'>
negative = <ufunc 'negative'>
not_equal = <ufunc 'not_equal'>
pi = 3.1415926535897931
power = <ufunc 'power'>
remainder = <ufunc 'remainder'>
right_shift = <ufunc 'right_shift'>
sin = <ufunc 'sin'>
sinh = <ufunc 'sinh'>
sqrt = <ufunc 'sqrt'>
subtract = <ufunc 'subtract'>
tan = <ufunc 'tan'>
tanh = <ufunc 'tanh'>
true_divide = <ufunc 'true_divide'>
typecodes = {'Character': 'c', 'Complex': 'FD', 'Float': 'fd', 'Integer': '1sil', 'UnsignedInteger': 'bwu'}
```